



Fill-in-the-Blank Applet

Field of the Invention

The present invention relates to a web-based learning environment. More
5 particularly, the present invention relates to an interactive software tool for testing a user
from a remote location.

Background of the Invention

The Internet has revolutionized educational opportunities by accommodating on-
10 line educational courses. On-line educational courses can provide a diverse educational
curriculum without imposing long-distance travel on students. Furthermore, on-line
educational courses accommodate students with a wide variety of lifestyles, provide
learning at an appropriate pace for a student and can accommodate a student's time
constraints. A student may enroll in a periodic on-line educational course to learn about
15 a new technology or other subject of interest without disrupting participation by the
student in other activities, such as a full-time job or activities with their family.

However, a drawback of current learning opportunities over the Internet is the
difficulty in testing a student. In the traditional classroom setting, testing has proven to
20 be extremely beneficial to the learning process of a student. Testing helps assess a
student's retention and absorption of presented information and allows an instructor to
chart a student's progress through the learning process. Testing also serves to reinforce
and emphasize the presented information in the mind of the student. Testing further
provides feedback to an instructor or course creator regarding the clarity and
25 comprehensiveness of the material presented. Without an effective testing process, a
distance-learning student may be merely a passive viewer of information, rather than an
active participant in the educational process.

Summary of the Invention

30 The present invention provides an interactive software application for testing a
user's knowledge in a web-based learning environment. According to an illustrative
embodiment, the present invention provides an interactive applet embedded in a course

page of a web-based curriculum. The applet provides a question to a user regarding the information presented in the course and instructs the user to provide an answer to the question. The applet provides feedback to the user, and allows the user to revise an incorrect answer.

5

According to an illustrative embodiment, the present invention provides a method comprising providing an interactive fill-in-the-blank applet, wherein the applet generates a Web page displaying a text box, a question and instructions to a user to enter an answer to the question in the text box.

10

According to an alternate embodiment, the present invention provides a method comprising receiving a request for a Web page and in response to said receiving step, sending a Web page. The Web page contains a question and a fill-in-the-blank applet embedded therein. The fill-in-the-blank applet displays instructions to a user to enter an answer to the question provided by the Web page.

15

According to another embodiment, the present invention provides a computer-readable medium for use in an electronic device. The computer-readable medium comprises instructions for running a fill-in-the-blank applet for displaying a question and a text box to a user, wherein the user can enter an answer to the question in the text box.

20

According to yet another embodiment, the present invention provides an electronic device comprising a processor, a display screen and memory. The memory of the electronic device includes a Web page having an interactive fill-in-the-blank applet embedded therein. The processor of the electronic device executes the fill-in-the-blank applet to generate a graphical user interface on the display screen. The graphical user interface displays a question, a text box and instructions to a user to enter an answer to the question in the text box.

25

30

Brief Description of the Drawings

Figure 1 is a block diagram of a distributed environment suitable for implementing an on-line educational course and running the fill-in-the-blank applet of an illustrative embodiment of the present invention.

Figure 2 is a block diagram of a client machine suitable for use in the illustrative
5 embodiment.

Figure 3 illustrates a web page including the fill-in-the-blank applet of the illustrative embodiment.

Figure 4 illustrates the fill-in-the-blank applet when a user enters a correct answer to the question asked by the applet.

10 Figure 5 illustrates the fill-in-the-blank applet after the user has incorrectly answered the question.

Figure 6 illustrates the fill-in-the-blank applet after the user has made the maximum attempts allowed to enter a correct answer.

15 Figure 7 illustrates the fill-in-the-blank applet after the user has entered an incomplete answer.

Figure 8 is a flow chart illustrating the steps involved in creating a fill-in-the-blank applet according to an illustrative embodiment of the present invention.

Detailed Description of a Preferred Embodiment

20 Figures 1 through 8, wherein like parts are designated by like reference numerals throughout, illustrate an example embodiment of a software application suitable for interacting with a user to test the user on information presented in an on-line educational course. Although the present invention will be described with reference to an illustrative embodiment shown in the figures, those skilled in the art will appreciate that
25 the present invention may be implemented in a number of different applications and embodiments and is not specifically limited in its application to the particular embodiment depicted herein.

The software application of the illustrative embodiment provides a mechanism
30 by which a provider of an on-line network learning center can test a student on information presented in an on-line education course setting of an on-line network learning center environment. An "on-line network learning center" is an organization

15

25

30

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99																																																																																																																																																												
0	00000000	00000001	00000010	00000011	00000100	00000101	00000110	00000111	00001000	00001001	00001010	00001011	00001100	00001101	00001110	00001111	00010000	00010001	00010010	00010011	00010100	00010101	00010110	00010111	00011000	00011001	00011010	00011011	00011100	00011101	00011110	00011111	00100000	00100001	00100010	00100011	00100100	00100101	00100110	00100111	00101000	00101001	00101010	00101011	00101100	00101101	00101110	00101111	00110000	00110001	00110010	00110011	00110100	00110101	00110110	00110111	00111000	00111001	00111010	00111011	00111100	00111101	00111110	00111111	01000000	01000001	01000010	01000011	01000100	01000101	01000110	01000111	01001000	01001001	01001010	01001011	01001100	01001101	01001110	01001111	01010000	01010001	01010010	01010011	01010100	01010101	01010110	01010111	01011000	01011001	01011010	01011011	01011100	01011101	01011110	01011111	01100000	01100001	01100010	01100011	01100100	01100101	01100110	01100111	01101000	01101001	01101010	01101011	01101100	01101101	01101110	01101111	01110000	01110001	01110010	01110011	01110100	01110101	01110110	01110111	01111000	01111001	01111010	01111011	01111100	01111101	01111110	01111111	10000000	10000001	10000010	10000011	10000100	10000101	10000110	10000111	10001000	10001001	10001010	10001011	10001100	10001101	10001110	10001111	10010000	10010001	10010010	10010011	10010100	10010101	10010110	10010111	10011000	10011001	10011010	10011011	10011100	10011101	10011110	10011111	10100000	10100001	10100010	10100011	10100100	10100101	10100110	10100111	10101000	10101001	10101010	10101011	10101100	10101101	10101110	10101111	10110000	10110001	10110010	10110011	10110100	10110101	10110110	10110111	10111000	10111001	10111010	10111011	10111100	10111101	10111110	10111111	11000000	11000001	11000010	11000011	11000100	11000101	11000110	11000111	11001000	11001001	11001010	11001011	11001100	11001101	11001110	11001111	11010000	11010001	11010010	11010011	11010100	11010101	11010110	11010111	11011000	11011001	11011010	11011011	11011100	11011101	11011110	11011111	11100000	11100001	11100010	11100011	11100100	11100101	11100110	11100111	11101000	11101001	11101010	11101011	11101100	11101101	11101110	11101111	11110000	11110001	11110010	11110011	11110100	11110101	11110110	11110111	11111000	11111001	11111010	11111011	11111100	11111101	11111110	11111111

JAVA is a trademark and registered trademark of Sun Microsystems, Inc. in the United States and other countries. As used herein, the term "applet" refers to a program designed to be executed from within another application. An applet is typically a small, specialized application written in the JAVA programming language that can be included

5 in an HTML or XML page, much in the same way an image is included and can be executed in a Web browser. Applets allow developers to add "interactive" content to Web documents (such as animation, page adornments, games, etc). Applets can be downloaded from a Web server and executed within a JAVA-compatible browser (e.g. HotJava) by copying code from the Web server to a client. JAVA source code files (i.e.

10 files with a *.java* extension) are compiled by a JAVA compiler to produce instructions in bytecode (i.e. files with a *.class* extension), which can then be executed by a JAVA virtual machine (VM). JAVA VM's are available for different platforms and thus, help to provide "platform independence" for JAVA programs.

15 To run a JAVA applet from a Web site, the site developer inserts applet tags (i.e. <applet>) into an HTML document, an XML document or the code of some type of page description language, which instruct the browser to download the appropriate classes from the Web server and then interpret the classes. "Classes" define JAVA program building blocks, called objects, as well as the behavior and attributes of the

20 objects. Specifically, the tag identifies the JAVA classes needed for running the applet and may also set parameter values for the applet.

The illustrative embodiment of the present invention is described relative to an applet in the JAVA programming language. However, the present invention is not

25 limited to the specific embodiment. It should be noted that other languages can have different forms of a program that perform the same function, and the present invention can be used in a programming environment other than JAVA. Furthermore, the present invention is not specifically limited to a JAVA applet executed from a Web browser, and may be implemented as various types of code modules in accordance with the

30 teachings of the present invention.

Figure 1 illustrates a block diagram of a distributed environment 100 suitable for implementing an on-line educational course and running the fill-in-the-blank applet of the illustrative embodiment of the present invention. The distributed environment 100 includes a Web server 101 and a plurality of client machines 102 connected to the Web server 101 and running a JAVA-enabled Web browser 103. The Web server 101 and the client machines 102 are connected to a network 100 via communications links 104. The Web server 101 includes a database 105, servlets 106 and Web pages 107 generated by the servlets 104. The Web browsers 103 on the client machines 102 locate and display Web pages. A Web browser further includes an HTML interpreter.

10

Figure 2 is a more detailed block diagram of an exemplary client machine 102. The client machine includes a central processing unit 109, a keyboard 110 enabling a user to enter data into the client machine, a video display screen 111 and a mouse 112 or other pointing device for controlling the movement of a cursor or pointer on the display screen 111. The client machine further comprises a network interface 113 for connecting the client machine to the Internet, a modem 114 for formatting data to be transmitted over telephone lines, coaxial cables or other suitable transmission lines, a decoder 115 for translates encoded data to its original format and client memory 116. The memory 116 contains Web pages 107 including embedded applets 117 or other software applications, a JAVA-enabled Web browser 103, and a JAVA class library 118 containing the JAVA classes needed to run the applets or other software applications. The client machine can be a PC, workstation, Internet appliance, cell phone, intelligent pager or other suitable device for running an on-line educational course.

25 *miss D47* To run the interactive fill-in-the-blank applet of the illustrative embodiment of the present invention, the Web server 101 sends an HTML document with an embedded fill-in-the-blank applet to a client machine 102 of the network 100. The HTML interpreter 108 then interprets the HTML document. The Java VM in the Web browser executes the fill-in-the-blank applet and displays a Web page on the video display screen 30 111 of the client machine 102. The GUI, which will be described in detail with reference to Figures 3 through 7, displays a question to a user and includes a box where the user can input an answer to the question.

105 D57 Figure 3 illustrates a Web page 10 generated by executing a fill-in-the-blank applet according to an illustrative embodiment of the present invention. The illustrated fill-in-the-blank applet tests a user regarding information presented in an on-line JAVA tutorial. In Figure 3, the interactive fill-in-the-blank applet is executed within the environment of a Netscape Navigator browser 11. The Web page 10 of the illustrative fill-in-the-blank applet displays question box 12 including a question related to information taught in an on-line course and instructions to the user. The Web page further includes a text box 13. The user is required to type an answer to the question in the text box. A label 15 directs the user to the text box 13. The Web page includes function buttons 19, 21 to facilitate interaction with the user. When a user clicks on a function button using a mouse or other pointing device, the applet responds according to particular instructions stored in an applet file.

15 In the Web page illustrated in Figure 3, an HTML document provides a static background for the applet, including the instructions and/or question to the user. For example, the HTML document includes a logo 14 and the question box 12 including the question to the user.

20 In Figure 4, the user completes typing in an answer and presses the "check" button 19 to assess his selections. If the user has entered a correct answer, the applet provides positive feedback in feedback site 20 to the user. According to the illustrative embodiment, the positive feedback in response to a correct match reads "Correct. That's the command we were looking for. Close this window to continue with the course."

25 According to the illustrative embodiment, the applet is contained in a "pop-up" web page that is accessed at a particular point in the on-line course, such as after the user finishes a lesson, or another applet. After the user completes the activity, the applet directs the user back to the on-line course.

30 However, if the user enters an incorrect answer, the applet provides a negative feedback in the feedback site 20 and prompts the user to try again, as illustrated in Figure 5. In the illustrative embodiment the negative feedback is "Incorrect. Try

again.” The fill-in-the-blank applet includes a “reset” button 21 to allow the user to clear his entry. After a predetermined number of failed attempts, the applet automatically provides the correct answer to the user, as illustrated in Figure 6.

According to the illustrative embodiment the correct answer to the question “What is the
5 command entered to compile JAVA code” is “javac”. According to the illustrative embodiment, after three attempts, the applet notifies the user “Your last response was incorrect. The correct response has been made for you. Please close this window to continue with the course.” The applet also disables the function buttons 19, 21 to prevent the user from subsequent action. According to the illustrative embodiment,
10 there may be more than one correct answer for a particular question. For example, the terms “java” “*Java” are also correct answers to the question recited in the question box 12.

According to one embodiment, shown in Figure 7, the applet further provides
15 feedback to the user regarding an incomplete answer. If the user fails to enter a word and then presses the “check” button 19, the applet responds “Please enter a valid response” or other similar command in the feedback site 20.

According to the illustrative embodiment of the present invention, the fill-in-the-
20 blank applet is embedded in a page within an on-line course that is sent to a client machine of the user from a server for the on-line network learning center. The page is an HTML, XML, or other page description language document including a link to the applet. An HTML file is a hypertext markup language file conventionally used in creating documents on the World Wide Web or Internet. HTML defines the layout and
25 structure of the file, using various tags and attributes. As discussed, the HTML file used in conjunction with the fill-in-the-blank applet of the present invention can include exam questions and reference the JAVA fill-in-the-blank applet that presents a question to the student, and then checks the answers.

30 An example of an HTML fragment suitable for implementation in the illustrative case includes the following code:

```

<html>
<body bgcolor="#ffffff" topmargin="0" leftmargin="0" marginwidth="0"
marginheight="0">
<table align="center" cellspacing="5" cellpadding="0" border="0">
5 <tr>
    <td>
    <table cellspacing="0" cellpadding="0" border="0">
    <tr>
        <td></td>
10 </tr>
    </table>
    <div align="center">
    <table width="98%" border="0" cellspacing="0" cellpadding="1"
bgcolor="#000000">
15 <tr>
    <td>
        <table border="0" cellspacing="0" cellpadding="1" width="100%">
        <tr>
            <td height="22">
20 <table width="100%" border="0" cellspacing="0"
cellpadding="5" bgcolor="#FFFFFF">
                <tr>
                    <td width="17%" valign="top" bgcolor="#000000"
align="center"><b><font color="#FFFFCC">Question:</font></b></td>
25 <td width="83%" bgcolor="#e4e4e4">What is the command
entered to compile Java code? Type the command and click the check
button. </td>
                </tr>
            </table>
30 </td>
        </tr>
        <tr bgcolor="#FFFFFF">
            <td height="22">
                <table width="100%" border="0" cellspacing="0"
35 cellpadding="5">
                    <tr><td>&nbsp;</td></tr>
                    <tr>
                        <td>
40 <div align="center">

```

Ans D67 This code defines the attributes of the HTML document wherein the fill-in-the-blank applet is embedded. Tag <td width="83%" bgcolor="#e4e4e4"> of the code precedes the question and instructions to the user displayed in question box 12 in Figure 3. The HTML code can be run with the help of a browser such as HotJava or Netscape Navigator. The questions can be viewed by any user running the HTML code using a browser by looking at the source code. Generally, a user can easily view the source code for a Web page by positioning a mouse pointer on the web page and clicking on the

right button of the mouse. According to the present invention, the answers to the question are included in a separate file and are not generally available to the user.

The HTML code further includes an applet tag. The applet tag instructs the browser to load an applet having the attributes defined by the tag. The browser calls a fill-in-the-blank applet of an illustrative embodiment of the present invention with the following applet tag HTML code:

```
<applet codebase="<prop servletname>content:/java/"  
10 archive="SESTBTInteractivity.jar" code="FillintheBlank" width="500"  
height="450"> <param name="file" value="fillintheblank.def"><param  
name="TBTServletBase" value="<prop servletname>"> <param  
name="TBTDocumentBase" value="<subst contentPageDocBase>"></applet>
```

The browser loads the applet class files for the fill-in-the-blank applet into a JAVA interpreter and calls a definition file defined by the applet tag and corresponding to the fill-in-the-blank applet. The tag tells the applet viewer or browser to load the applet whose compiled code is in the file "FillintheBlank" and sets the initial size of the applet to 500 pixels in length and 450 pixels in height. The "codebase" parameter specifies the base URL of the applet (the directory that contains the applet's code). The "archive" parameter of the tag describes one or more archives containing classes and other resources for the fill-in-the-blank applet. The .jar files referenced in the archive are compressed versions of JAVA files, suitable for transporting over a network, such as the Internet. The "param name" parameters specify an applet-specific attribute. The first param name invokes a definition file for the fill-in-the-blank applet. The second param name parameter, TBTServletBase, specifies the server for the applet and the parameter TBTDocumentBase specifies the directory containing the SESTBTInteractivity.jar file.

The JAVA definition file written for the fill-in-the-blank applet of the present invention includes the answers and other parameters of the applet, which are hidden from the user. The applet tag directs the browser to the definition file "fillintheblank.def" stored on the server. The definition file defines the appearance of the applet and includes a number of attribute tags. For example, a definition file for the fill-in-the-blank applet of the illustrative embodiment the present invention may include

the following code, which is illustrative of one embodiment of the invention and is not to be construed in a limiting sense:

```
5  @bgcolor
   white

   @rows
   1
10  @columns
   15

   @scrollbars
15  none

   @setlabel
   Answer:

20  @correct
   @exactmatch
   javac

   @correct
25  @exactmatch
   Java

   @correct
30  *Java

   @correctfeedback
   Correct. That's the command we were looking for.   Close this window to
   continue with the course.

35  @incorrectfeedback
   Incorrect. Try again.
```

The definition file tags define particular parameters of the applet. In the example set forth above, "bgcolor" defines the color to be displayed as a background. As defined by the illustrative definition file of the fill-in-the-blank applet, the background color is set to white. The tag "rows" defines the number of rows for the text field and the tag "columns" defines the number of columns in the text field. In the illustrative example, the fill-in-the-blank applet includes one row and fifteen columns for the text field. The tag "scrollbars" defines the number of scrollbars visible in the fill-in-the-blank applet. According to the illustrative embodiment, no scrollbars are visible. The tag "setlabel" defines the text before the text field. In the illustrative example, the label is "Answer:". The tag "correct" defines the text to be associated with a correct response. According to

the illustrative embodiment, an "exactmatch" tag follows two of the "correct" tags. The "exactmatch" tag requires that the response entered by the user exactly match the text in the "correct" tag in matters of case sensitivity. According to the illustrative embodiment, the fill-in-the-blank applet requires that the user either enter "javac" or
5 "Java" with the precise case sensitivity, or "*Java", without having to exactly matching the text. The tag "correctfeedback" defines the feedback to be given to the user when the user enters a correct answer. Finally, the tag "incorrectfeedback" defines the feedback to be given to the user when the user enters an incorrect answer.

10 The definition file may further include a tag defining the maximum number of attempts a student can make before the applet automatically provides the correct answer. In general, an exam may include a several questions, and the maximum number of attempts defined by the applet can vary according to the difficulty of each question. For example, an exam may include two questions, the first of which may be deemed by the
15 administrator to be easier than the second. In such a case, the maximum attempts may be set to a smaller number for the first question than the second question. This provides the student more opportunities to obtain the right answer for the second, more difficult question.

20 One advantage to providing the answers to the exam question in a separate file, instead of the HTML source file that references the JAVA applet, is that a user cannot cheat by looking for the answers to a question in the source file, which is usually freely readable. Second, the JAVA applet requires the correct answer, to check whether or not a student's reply is accurate, but does not require the question. Thus, it is convenient to
25 utilize a separate file as input to the JAVA applet that contains answers but no questions. The definition file can be altered, and subsequently used by the JAVA applet, without having to recompile or rejar the applet. Hence, the use of a definition file facilitates the introduction of new questions and the correction of errors. For example, after the applet executes, it may become evident that there is an error, such as a misspelled word in a
30 message provided to the student. The misspelled word can be modified directly in the definition file, and the JAVA code run again, without having to recompile and/or rejar the applet.

Figure 8 illustrates the steps involved for an on-line educational course developer to create the fill-in-the-blank applet of the illustrative embodiment of the present invention. At some point in the educational course, the developer may include the fill-in-the-blank applet of the present invention to test the user's knowledge of the presented material. First, in step 80, the developer embeds a snippet in a course page of the on-line educational course for calling the fill-in-the-blank applet. The snippet directs the browser and the user to an HTML document containing the fill-in-the-blank applet. For example, a suitable code fragment in a course page for calling a pop-up HTML page containing the applet according to the teachings of the present inventions is as follows:

```
<a href="HOTTEXT?DialogURL=self-check/self-  
check.html&DialogWidth=640&DialogHeight=480&DialogTitle=Self-  
Check&pgtype=framehotttext"></a>
```

The above code allows a user to launch a pop-up page containing an examination question. The pop-up page of the illustrative embodiment has a width of 640 pixels and a height of 480 pixels.

20

Next, in step 81, the developer populates the question text and builds the pop-up page for the applet. The pop-up page coding includes the above-described applet tag or another suitable tag for executing a fill-in-the-blank applet according to the teachings of the present invention. According to a preferred practice of the invention of the invention, the developer sets the width dimension of applet in the applet tag such that the function buttons (i.e. the check button and the reset button) are visible.

25

The developer creates a fill-in-the-blank definition file in step 82, including the desired parameters and tags, as described above. Finally, in step 83, the developer tests the functionality and desired presentation of the fill-in-the-blank applet within an on-line educational course.

30

To implement the fill-in-the-blank applet of the present invention, the fill-in-the-blank applet includes computer executable instructions stored on a computer readable

medium. The medium can include a hard disk, RAM medium CD-ROM, or diskette, for example. The instructions can be stored on a server that can be remote from the user. To run the application, the user can download the instructions to a computer readable medium of a local computer. The instructions can then be downloaded from the
5 computer readable medium of the local computer to a local processor of the local computer, where the instructions are executed with the help of a virtual machine. A graphical user interface is generated by the instructions for displaying a question and a text field. The interface allows a user to enter an answer to the question in the text field.

10 Although many of the foregoing examples have highlighted the use of JAVA in the present invention, other computer languages executed by virtual machines may be utilized for this purpose. As known to those of ordinary skill in the art, a virtual machine is an abstract computing machine having an instruction set and being capable of manipulating various memory areas at run time. A well-known virtual machine is the P-
15 Code machine of UCSD Pascal, and the JAVA virtual machine. The JAVA virtual machine does not assume any particular implementation technology, host hardware, or host operating system. It is not necessarily interpreted, and can be implemented by compiling its instruction set to that of a silicon processor. The JAVA virtual machine may also be implemented in microcode or directly in silicon.

20 These examples are meant to be illustrative and not limiting. The present invention has been described by way of example, and modifications and variations of the exemplary embodiments will suggest themselves to skilled artisans in this field without departing from the spirit of the invention. For example, the present invention is
25 not limited to a JAVA applet executed from a Web browser, and can include any program executed from another application. The described applet is not limited to execution from a pop-up Web page and can be embedded in any suitable document code written in any suitable programming language. Alternately, the question and/or instructions to the user can be part of a definition file for the applet, rather than a part of
30 a static background created by a Web page or other suitable document. Features and characteristics of the above-described embodiments may be used in combination. This description, including the written code, is to be construed as illustrative only and is for

10

[illegible]